Peptide Docking Using Dynamic Programming

KAMALAKAR GULUKOTA, SANDOR VAJDA, and CHARLES DELISI*

Department of Biomedical Engineering, Boston University, Boston, Massachusetts 02215

Received 19 April 1995; accepted 28 August 1995

ABSTRACT

An extended dynamic programming algorithm is presented that is applicable to the fragment assembly phase of the *site mapping fragment assembly* approach to peptide docking. After constructing a free energy map of the receptor using each of the amino acids in the peptides to be docked, we apply the algorithm to two systems: HIV-1 protease complexed with a synthetic hexameric inhibitor, and MHC HLA-A2 complexed with a nonameric peptide. The all atom root mean square deviation between the predicted and crystal structures was 1.7 and 2.0 Å, respectively. While these results are reasonable considering the relatively coarse level of mapping, the more important result is that the structures are probably very close to the best obtainable by an exhaustive search through the entire data map, and yet are obtained with a reduction of 3–5 orders of magnitude in the number of computations. We also outline a prescription for an iterative procedure which finds the global minimum with increasing confidence.

Introduction

Drug design strategies invariably require the screening of large numbers of potential ligands for specificity and affinity for a given protein target. Computational methods for rapidly and reliably rank ordering candidates would evidently be an important component of such strategies. A useful predictive procedure must address two major difficulties: the development of an accurate method for rapidly evaluating free energy differences between structures, and the development of a rapid algorithm for searching the complex free energy landscape of possible structures for a global minimum.

Ab initio free energy calculations have a long history in the chemical literature, and a number of methods, including free energy perturbation and thermodynamic integration,[1] are in principle available for approaching that part of the problem. However, because their use in docking procedures is not computationally viable, a number of investi-

*Author to whom all correspondence should be addressed.

CCC 0192-8651 / 96 / 040418-11

gators have used alternative, relatively simple, rapidly evaluatable, but incomplete target functions (see ref. 3 for a review). These include geometric complementarity and electrostatic interaction energy[4-7] and solvation and entropic components of the free energy.[8,9] In this work, a semiempirical free energy function is used that is easily evaluated and reasonably complete.[10]

In general the most effective methods for approaching the other major problem in docking, finding the bound conformation of short flexible ligands, are variants of a procedure first introduced by Goodford,[11] and consist of "building" the ligand from its component functional groups within the binding site of the protein. The first step in this approach is to determine an *energy map* of the site of the protein using the various functional groups in the ligand as probes. Such maps have not been constructed to date using free energy as a target function and would, in their most general representation, sample the orientational, conformational, and translational positions of the probe, taking account of local conformational changes in the binding site at each probe location. The map can be constructed in various ways, for example, by sampling conformations formed by perturbing a known minimum,[12] by multiple copy simultaneous search (MCSS),[13,14] by a grid search,[11,15] and by building the ligand with functional groups from a predefined library.[16]

In the case of a peptide, the result of such *site mapping* would be a list of favorable conformations, orientations, and C^{α} positions for each amino acid residue. Favorable conformations for a given amino acid are those exceeding its deepest free energy minimum by at most a cutoff parameter. All conformations beyond the cutoff are rejected as unfavorable. In general, however, a conformation that is unfavorable for an individual amino acid might be favored in the complex due to interactions with other residues in the peptide. In choosing only those within a cutoff, we are implicitly assuming that the approximate value of the target function for the globally optimum structure can be obtained by summing (adding) contributions to the target function from its fragments. That is to say, the target function of the whole peptide is nearly additive over contributions from its fragments.

After site mapping, the peptide is built by a *fragment assembly* algorithm that selects the states of the sequence of amino acid residues in a way that optimizes the overall free energy. We report on a new method, based on dynamic programming, for addressing this concatenation problem and apply it to a data map generated using a recently introduced free energy function.[10] The extended dynamic programming methodology and its limitations are explained and its use in peptide docking is described and applied to two receptor-ligand systems: the human leukocyte antigen (HLA)-A2 peptide system analyzed by Sezerman et al.[17] and the human immunodeficiency virus (HIV) protease inhibitor system analyzed by King et al.[12] Then we present results indicating all atom root mean square deviations (rmsds) between observed ligand structures and those predicted to be within 5 kcal/mol of the minimum. These rmsd's range between 1.5 and 2.2 Å, with global minima of the target function corresponding to structures with rmsds of 1.7 and 2 Å, respectively. Finally, we analyze the methodology further and discuss the requirements for more accurate predictions.

Principle of Optimality and Dynamic Programming

Fragment assembly involves concatenating amino acid conformations, orientations, and locations so that the combination optimizes the target function of the system. This can be posed as a general combinatorial optimization problem (Fig. 1). In the most general case, the only method that guarantees finding the optimal combination is an exhaustive search through all possible combinations. Combinatorial explosion precludes the use of this method for all but very small problems.[18] For larger problems, dynamic programming[19] provides an efficient solution if the so-called principle of optimality (Fig. 1) is satisfied. This principle, a reformulation of the near additivity criterion mentioned above, can be explained in the following manner. For concreteness, let the solid line **a2–b3 ··· g3–h2** shown in Figure 1 be the optimal path between nodes **a** and **h**. For ease of reference we will refer to the optimum value of a target function over a set of *subpaths* having identical terminal nodes, as a local optimum. The principle of optimality states that the target function evaluated on any subpath contained in the globally optimum path must have a locally optimum value. For example, the subpath **d1–g3** is the locally optimum path connecting **d1** with **g3**. The principle is obviously true if the target function is additive over subpaths (i.e., if its global optimum can be written as the sum of local optima). Thus if the

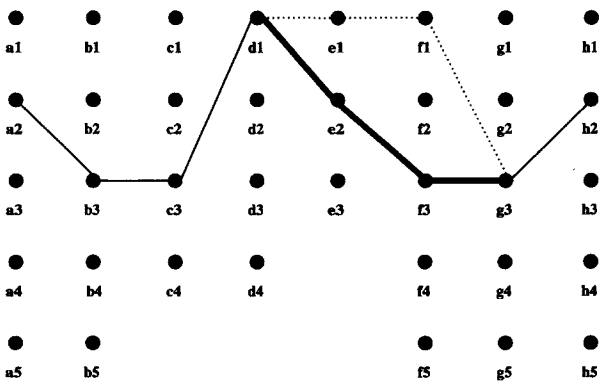

FIGURE 1. The first stage of *site mapping–fragment assembly* is mapping the binding site. This gives rise to a list of possible conformations for each residue. Shown here is a schematic diagram with residues running from **a**–**h**. Each residue has listed under it the different conformations it can be in. Thus **a** has five conformations **a1** ··· **a5** and similarly for the others. Finding the structure of the peptide is equivalent to creating a directed path in the figure, for example, **ai bj ck**.... Thus the assembly problem is a case of combinatorial optimization. Principle of optimality can be explained in the context of this figure. Assume that the bold line **a2** ··· **h2** is a globally optimal path. The principle of optimality states that any part of an optimal path is itself optimal. For example, **d1 e2 f3 g3** is the optimal path connecting **d1** and **g3**. Any other path such as the dotted line is not optimal. When a problem obeys the principle of optimality, dynamic programming is a very efficient search algorithm.

TABLE I.
Classical Dynamic Programming Algorithm.

```
 1  LastKept = Empty;
    Total = Number of Nodes;
 3  KeptPositions = All positions of node 1;
    Loop (I from 2 to Total)
 5      { AllPoss = All positions of node I;
        Loop (J over AllPoss)
            { NewKept = Empty;
            EnergyList = Empty;
            AllSP = Empty;
10          Loop (K over KeptPositions)
                { CurrentSubpath = Combine (K, J);
                Ener = Energy (CurrentSubpath);
                Add Ener to EnergyList;
                Add CurrentSubpath to AllSP;
                }
16          If (This is Last Residue)
                Add AllSP to LastKept;
            Else
                { MinEner = Minimum of EnergyList;
20              MinPath = Path of MinEner;
                Add MinPath to NewKept;
                }
          }
        KeptPositions = NewKept;
      }
26  OptimalPath = Optimum among LastKept;
```

dotted line, rather than the heavy line, between **d1** and **g3** were a local optimum, we could replace the heavy line with the dotted line. If the principle of optimality prevailed, this would lead to a different global optimum, and recover the situation in which each subpath on the global optimum is locally optimal.

If the system obeys the principle of optimality, a dynamic programming algorithm can be used to build an optimal path. This algorithm, which is written out in Table I, can best be explained recursively (Fig. 2). Assume that the peptide has been built up to the $(i-1)$th residue from the the N-terminus and suppose the $(i-1)$mer has S_{i-1} conformations. The next stage in the buildup involves adding the next residue to obtain $i$mers. This is done as follows.

The first conformation of the ith residue is attached to all $(i-1)$mers thus forming S_{i-1} $i$mers. The target function is evaluated for each of these $i$mers, and only the conformation corresponding to its minimum is retained. This procedure is repeated with the other conformations of the ith residue so that we finally obtain one $i$mer for each of the S_i conformations of the ith residue.

In the next buildup stage, each of the S_{i+1} conformations of the $(i+1)$th residue is combined with all $i$mers. This procedure is continued until the end of the chain is reached. This recursive process is seeded with "monopeptides" i.e., conformations of the first residue (step 3 in Table I). The most important point to remember here is that, in the ith stage of the buildup, the comparison of the target function is always done within a set of partially built peptides whose last residue is in the same conformation (Fig. 2). This is a consequence of the principle of optimality and the essence of the forward dynamic programming algorithm.

This algorithm is extremely efficient compared to an exhaustive search. For a system with a total of N nodes and S positions at each node, an exhaustive search requires $O(S^N)$ evaluations of the target function, while dynamic programming requires $O(NS^2)$ evaluations.

For systems that obey the principle of optimality only approximately, we can use extended dy-

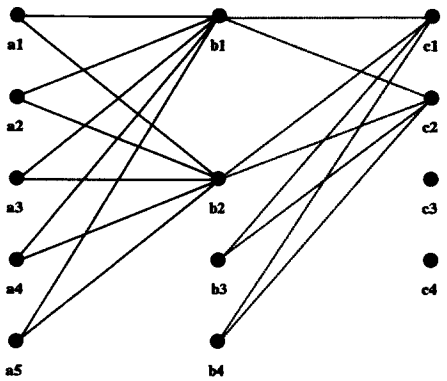

FIGURE 2. Dynamic programming algorithm builds the optimal path one step at a time. At stage **a** in this schematic, there are five possibilities. All of these are combined with each of the possibilities for stage **b**. Specifically, **a1 – a5** are combined with **b1**. The target function is evaluated for each of these five paths and only the minimum is retained. The same is done for all the other possibilities of stage **b** (i.e., **b2, b3,** ...). Thus, at the end of this stage of buildup, there will be as many possibilities of paths *up to* **b** as there are possibilities at **b**. The next buildup step combines all of these paths with each of **c1, c2** .... An important point to make here is that minimum at every buildup stage is determined only among those paths that end in the same position. For example **a1 – b1, a2 – b1,** ..., **a5 – b1** are compared only among themselves. In particular they are not compared to **a1 – b2, a2 – b2,** ..., **a5 – b2**. This latter set, once again, is only compared within itself. This is the essence of the forward dynamic programming procedure.

namic programming (EDP)[20] to obtain the optimal path relatively efficiently. In this, we alter statement 20 in Table I by defining a cutoff value. MinPath is then defined as the set of *all* subpaths which are within a cutoff of the minimum (MinEner). Clearly, as the cutoff increases the algorithm approaches an exhaustive search. Typically, the performance deteriorates precipitously with increments in the cutoff value.

TARGET FUNCTION

The target function we use is an empirical free energy function formulated recently by Vajda et al.[10] Briefly, ΔG, the free energy change on binding is

$$\Delta G = \text{const} + E_{\text{rl}}^{\text{el}} + \langle \Delta G_{\text{h}} \rangle + \langle \Delta E_{\text{l}} \rangle - T\Delta S_{\text{sc}} - T\Delta S_{\text{bb}}. \quad (1)$$

Here const is a number that is independent of the details of the interaction, and reflects the loss of rotational, vibrational, and cratic free energies. $E_{\text{rl}}^{\text{el}}$ is the electrostatic interaction energy between the receptor and the ligand, $T\Delta S$s are the contributions to the free energy from the loss of side chain (sc) and backbone (bb) entropy, and $\langle \Delta E_{\text{l}} \rangle$ is the difference in the internal energy of the peptide between its bound and free states. $\langle \Delta G_{\text{h}} \rangle$ represents the hydrophobic contribution to the binding free energy and is defined as the difference between the hydrophobic energies of the complex (rl), receptor (r), and the ligand (l).

$$\langle \Delta G_{\text{h}} \rangle = \Delta G_{\text{h}}^{\text{rl}} - \langle \Delta G_{\text{h}}^{\text{r}} \rangle - \langle \Delta G_{\text{h}}^{\text{l}} \rangle$$

where $\langle \Delta G_{\text{h}}^{\text{r}} \rangle$ and $\langle \Delta G_{\text{h}}^{\text{l}} \rangle$ are Boltzmann weighted averages over all the free states because the free state is not uniquely defined. It may be noted that the function treats the ligand in greater detail than it does the receptor. For example, while changes in receptor side chain entropy and hydrophobicity are calculated, the changes in its internal energy are not explicitly included in the target function. The assumption is that the receptor conformation does not change as drastically as ligand conformation.

When the peptide sequence is specified the problem is one of docking rather than design, and we can ignore the free state as well as the constant in eq. (1). The function then is not a free energy, but just a target function to be optimized. For example, $T\Delta S_{\text{bb}}$, $\langle \Delta G_{\text{h}}^{\text{r}} \rangle$, and $\langle \Delta G_{\text{h}}^{\text{l}} \rangle$ are constants as long as the sequence remains the same and they can therefore be ignored. Also, $\langle \Delta E_{\text{l}} \rangle$ is the Boltzmann weighted average difference between the energy of the bound and free peptide. The energy of the free peptide, however, is constant as long as the peptide sequence does not change. Hence we can substitute for this average a single variable E_{l} which is the internal energy of the peptide in the bound state. Thus our target function is:

$$\Phi = E_{\text{rl}}^{\text{el}} + \Delta G_{\text{h}}^{\text{rl}} - T\Delta S_{\text{sc}} + E_{\text{l}}. \quad (2)$$

An assumption was made in formulating the above free energy [eq. (1)] function in ref. 10, namely the total van der Waals energy is left unchanged by complex formation. The underlying physical picture, which is common in the literature,[8, 21, 22] is that ligand–protein van der Waals interactions that are lost upon dissociation are balanced by van der Waals interactions with solvent. The free energy is therefore mapped on a space in which van der Waals clashes have been removed.

In practical terms, this means that all van der Waals clashes must first be removed before the target function can be applied to a predicted conformation.

PEPTIDE DOCKING WITH EDP

A principal reason that dynamic programming can fail to find the global minimum in protein folding problems is that the minimum free energy structure of a fragment (during buildup) may not be the same as the structure of this fragment in the native protein. This can happen when distantly separated portions of the chain can interact more favorably with one another when they are locally nonoptimal rather than when they are locally optimal. EDP attempts to overcome this difficulty by retaining more than one state per segment. This procedure works efficiently only if the locally optimal values of the target function (i.e., the values assumed on the individual segments) *almost* add up to the globally optimum value (i.e., the value on the entire peptide). Another way of saying this is that each locally optimum value must not be too far from the value of the local component of the globally optimum target function. In the rest of this article we call this the near-additivity criterion.

This criterion is likely to be satisfied in peptide docking because the dominance of the peptide–protein interaction forces a relatively extended structure on the peptide, thereby eliminating the possibility that distant segments of the peptide can interact strongly with one another. Thus, the one term in eq. (2) which could cause deviation from additivity, namely the internal energy, will not present a problem for the systems of interest in this study, or for any similar systems in which the bound ligand conformation precludes direct interactions between distant segments.

Nearby segments might of course interact most favorably when they are in nonoptimal conformations, to the extent that the optimum for the entire segment might result from two fragments in individually nonoptimal conformations. However, so long as the system is nearly additive, this difficulty is efficiently overcome by using a small cutoff parameter.

We therefore expect that EDP, which is reliable for peptides in solution, should be even more suitable for bound peptides, and should allow the choice of a relatively small cutoff parameter. It is also possible to iterate over cutoff values, as explained later, to determine the global minimum with increasing confidence.

Due to the assumption made in the free energy function about the cancellation of van der Waals interactions, any predicted conformation must be energy minimized so that any large van der Waals clashes are mitigated. The slight movements of atoms that occur during this minimization form another potential source of minor deviation from additivity of the target function. Thus there are two potential sources of nonadditivity in the target function—the minimization process and the internal energy of the peptide. Both are expected to be minor for a variety of complexes, including those considered here.

We use EDP to predict the docked structure of a peptide inhibitor to HIV-1 protease and of an influenza peptide to the major histocompatibility complex (MHC) HLA-A2. Different mapping strategies were used in each case. In the following we briefly describe these methods for generating what in effect is the raw data for the algorithm, describe the results of constructing the conformation of the bound peptides by dynamic programming, and discuss the relative merits and drawbacks of the algorithm.

Methods

MAPPING

In its most general form, the input to the mapping routine is the structure of the binding site of the protein (e.g., the coordinates of the backbone atoms and the conserved side chains) and the sequence of the peptide. For each peptide residue, the output consists of a list of conformations compatible with the binding site (which is treated as locally adaptive) and their binding free energies. Thus, mapping is a description of the binding site and its variations about a major structural theme, rather than that of the peptide. These data are then used as input into some buildup procedure (EDP) to construct low free energy conformations of the peptide. The mapping results presented below are based on two different mapping methods: one used by King et al.[12] for docking an inhibitor to HIV protease, and the other by Sezerman et al.[17] for docking MHC binding peptides. The two examples have different mapping densities: the protease system is mapped relatively densely over a small portion of the conformation space and the

MHC system is mapped relatively sparsely, but samples a larger portion of the conformation space.

Mapping Protease System

In the case of the HIV-1 protease we consider the inhibitor peptide MVT101 which has the sequence *N*-acetyl-Thr-Ile-Nle-Ψ[CH_2—NH]-Nle-Gln-Aar (Nle, norleucine and Aar, arginine amide).[23] The map was constructed as follows.[12] The crystal structure of complexes with a large number of different inhibitors, including MVT101,[23] are known. The structures of the different inhibitors are very similar to each other, a fact that is utilized by taking an average consensus structure as a starting point. The map is then developed by perturbing this consensus structure to extensively explore a small part of the conformational space about it. The perturbation consists of translating the C^α of each residue by 0.3 Å along all three axes in both directions. Similarly the orientation of the C^β with respect to C^α was perturbed by rotating the angle by 30° along all three axes, giving seven rotational positions for each of the seven translational positions. Thus free energies were calculated for 49 different positions for each rotamer of each side chain, generally resulting in several hundred possible states for each side chain. Since the total number of combinations is the product of the number of states for each side chain, the size of the total space is well beyond what is amenable for exhaustive exploration.

Mapping MHC System

The crystal structures of complexes of five different peptides bound to the HLA-A2 class I molecule were determined.[24] The peptides have highly homologous backbone structures, especially toward their amino and carboxy termini, with variability increasing to about 3 Å for the central α carbons. Variability in side chain orientations exhibits similar regularity: conservation is strong for the terminal residues and is lost almost entirely for the central side chains.

As a second application of the algorithm, we determine how effectively the observed side chain orientations can be predicted. Directional prediction is especially important for planning experiments since orientation determines whether a side chain interacts with a T cell or an MHC molecule.

Since the predictive goal is reasonably coarse, the mapping is sparse, although there would be no difficulty in principle in increasing its resolution. Sezerman et al.[17] started with the structure of the HLA-A2/HIV-1 GP120 peptide (TLTSCNTSV) complex[24] and did an orientational and conformational search of each side chain. Using C^α coordinates from this complex, side chains of the test sequence (influenza peptide GILGFVFTL) were rotated through 360° in 30° intervals. At each rotation, an exhaustive search was done of the possible side chain conformations using the program CONGEN.[25] The conformation that gave the lowest energy (as calculated by CONGEN) was preserved and the others discarded, leaving 12 conformations for each residue.

BUILDUP PROCEDURE

The maps are stored in the form of coordinate files, with each valid position defined by the coordinates of all its atoms. Some of these valid positions can be rejected because they are too high in energy. This is done by defining a *list cut* value E_{lc}. The minimum free energy position of each residue is determined. Then all positions whose free energy exceeds this minimum E_{min} by more than the list cut value are rejected. $E_{lc} = 0$ restricts consideration to the minimum energy conformation. As the value increases, more and more of the mapped positions are considered until *all* the mapped positions are accepted when list cut is the free energy difference between the minimum and the maximum free energy.

For the protease system we defined an array of list cut values (Table II). These values can be changed easily if desired. In the case of the MHC system we use a list cut value equal to the maximum difference in target function, i.e., we use all 12 mapped positions of every residue. However, it must be remembered that at each of the 12 posi-

TABLE II.
Combinatorial Explosion from Mapping Data for Protease System.

Residue Name	List Cut (kcal / mol)	No. conformations	
		Mapped	Chosen
1 Ath	1.0	98	8
2 Ile	0.5	147	10
3 Nle	0.5	147	5
4 Nle	0.5	147	15
5 Gln	0.5	294	7
6 Aar	0.5	245	4
Total combinations		2.2×10^{13}	168,000

tions, we have already rejected a large number of side chain conformations and retained only the conformation with the minimum CONGEN energy. It is worth pointing out once again that in rejecting structures of large energies, we are implicitly assuming near additivity in locally optimum values of the target function.

After thus choosing the conformations for the individual residues, they are assembled into peptide fragments by concatenating their coordinate files using the Unix cat program. Every such fragment is then energy minimized using an adopted basis Newton–Raphson (ABNR) method for 120 steps for the protease system and 160 for the MHC system. The target function is evaluated for this minimized fragment. [Target function of eq. (2), rather than the full free energy function of eq. (1), is used since we are interested in docking rather than design.] The target function values of the built fragments are then compared. In accordance with the EDP algorithm, the comparison in the case of the partially built fragments is done only between fragments that end in the same conformation of the same residue.

Results

The results from dynamic programming should be examined with reference to two main issues. First, how close is the minimum ascertained by dynamic programming to the global minimum? The global minimum here refers to the minimum of all combinations in the mapping data. Second, how well does this minimum relate to the crystal structure? This question is really about the adequacy of the resolution of the map and the accuracy of the target function rather than about the algorithm.

PROTEASE–INHIBITOR SYSTEM

The first question cannot be answered with certainty unless the true global minimum of the target function is ascertained through an exhaustive search of all possible combinations contained in mapping. Table II shows the number of these possible combinations for the protease–inhibitor system. Even after choosing only a small number of conformations out of the total number mapped, the number of combinations is 168,000. Since each evaluation of the target function has to be preceded by a minimization, this requires 168,000 calls each to the minimization and the target function routines. Comparison among these 168,000 target function values will then give the global minimum.

Our algorithm reduces the complexity by about 3 orders of magnitude and estimates the global minimum with only 395 calls each to the minimization and the target function routines. The main source of this speedup is the fact that at every step, the algorithm carefully evaluates and chooses only the "good" conformations for the partial peptide ending in the current residue. Thus, all but 40 of our 395 calls refer to partially built peptides. Then, the possible positions of the next residue are combined only with these chosen conformations, converting an exponential search into a polynomial one. The complexity of the algorithm is analyzed in greater detail in the Discussion section. Forty of the 395 calls refer to the full peptide, i.e., our effort gives rise to 40 different structures for the full peptide which are presumably close to the global minimum.

To investigate how close our low energy structures are to the true global minimum, we calculate their rmsds from the crystal structure. Figure 3A shows the rmsds plotted against free energies for some of the structures built. The structure of the lowest free energy has an rmsd of 1.66 Å. All 40 structures are within 2.03 Å of the crystal. The calculation of the target function has a resolution of about 5 kcal/mol. That is, minor variations in structure can lead to changes in the target function of about 5 kcal/mol. Eleven of the 40 structures are within 5 kcal/mol of the minimum; their rmsds range from 1.54 to 1.82 Å. Finally, the prediction closest to the crystal structure has an rmsd of 1.54 Å and is ranked second best in energy; its energy exceeds the minimum by 1 kcal/mol.

HLA-A2: INFLUENZA PEPTIDE SYSTEM

The sequence of the influenza peptide used (GILGFVFTL) has glycines at positions 1 and 4. Since the mapping in this system was one of side chain rotation, and since glycine has no side chain, no mapping was done on these residues. So, for these two residues, the mapping data base contains unique positions corresponding to their configurations in the HIV-1 GP peptide (the starting point for the mapping). The other seven have 12 possible positions each. Since our list cut is infinite, the total number of combinations in this system is, $12^7 = 3.58 \times 10^7$. So an exhaustive search requires 3.58×10^7 calls each to the minimization

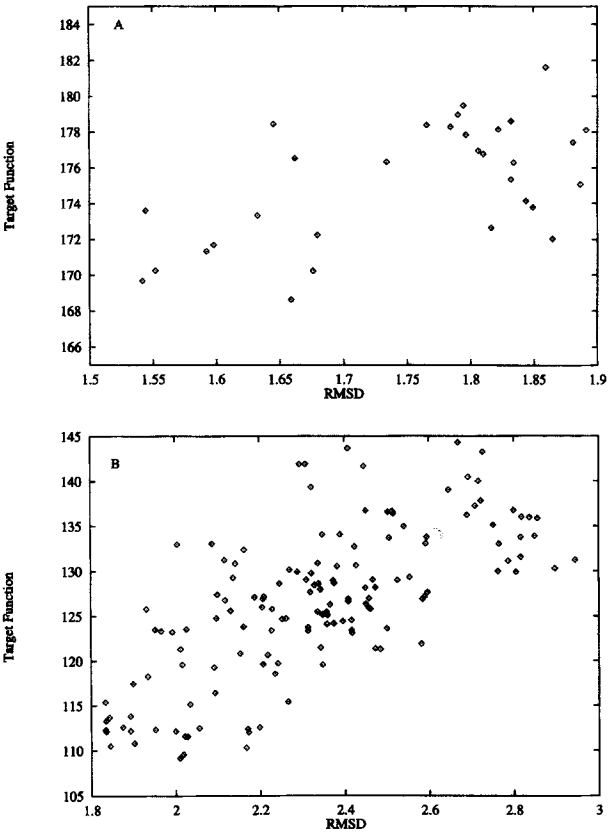

FIGURE 3. Target function of predicted structures plotted against the all atom rmsd from the crystal structure. (A) Structures predicted for the protease inhibitor system. Some high energy predictions have been deleted for clarity. (B) Structures predicted for the influenza peptide system. Twelve predictions with relatively high energy and high rmsd are deleted for clarity.

and the target function routines. By contrast, our algorithm involves 818 calls. Once again, this 5 order of magnitude reduction is because the algorithm is polynomial while exhaustive search is exponential. The effort finally gives rise to 156 fully builtup structures. Figure 3B shows the free energies of these structures plotted against their rmsds. The lowest energy corresponds to an rmsd of 2.01 Å from the crystal. All but 12 of the 156 structures are within 2.95 Å of the crystal. There are 20 predictions within 5 kcal/mol of the minimum, ranging in rmsd from 1.83 to 2.19 Å. The prediction closest to the crystal structure is 1.83 Å from the crystal. Its target function is ranked 12th best and is 3.1 kcal/mol from the minimum. As explained in greater detail in the Discussion section, improved quality structures would probably result if the fineness of the map was increased.

FOLLOWING BUILDUP

The cutoff parameter is the single most important determinant of the efficiency of the algorithm. It has to be small enough that the procedure does not degenerate into an exhaustive search. On the other hand, it has to be large enough that no intermediate structure which can potentially lead to the global minimum is rejected prematurely. It was set arbitrarily at 0.2 kcal/mol. An estimate of how good this value is can be made by looking at the free energy values of all the intermediate structures generated, irrespective of whether they were accepted or rejected.[26] Our program enables us to do this by generating a list of all intermediates and their energies. This list contains one line for every call to the free energy function.

At each buildup stage, all structures at that stage are compared. This comparison is between those that are all of the same length and have the same conformations for their last residue (Fig. 2). So, one can split the list into subsets such that all members of a subset have the same length and the same conformation for their last residue. We construct these subsets and determine the minimum free energy value in each. Then, we determine the rejection and acceptance margins of all the other members of the subset. It is important that a large number of structures not be rejected very close to the cutoff barrier. This could be a sign that a large part of potentially relevant mapped space is being ignored. One way to deal with this is to increase the cutoff parameter to include these marginal rejects. In both the systems examined, the large majority of the rejections are due to energies that are much larger than the cutoff barrier, showing an internal consistency within the method.

Discussion

The quality of the predicted structures and the efficiency of the procedure depend on the adequacy of the map, the validity and speed of concatenation, and the accuracy of the target function. In this article we have focused on concatenation, and in particular on a new approach based on dynamic programming.

The range of validity of dynamic programming in the context of docking can be formally understood as follows. Consider many conformational states $1, 2, \ldots$ representing the possible structures of the partial peptide ending one residue before A (Fig. 4). Let B be the conformation of the rest of

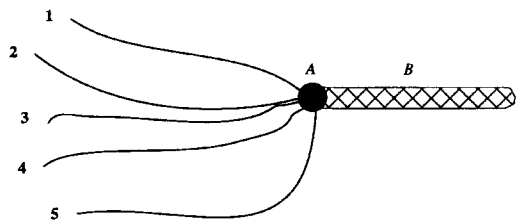

FIGURE 4. A theoretical determination of the cutoff parameter. Consider five partial peptides. When the residue *A* and the partial peptide *B* are added on to any of the partial peptides (1 ··· 5), a full peptide results. At this stage, the decision being made is which of the five partial peptides should *A* be built on to. The total energy of all the fully built peptides is shown in eq. (3). If five is the *minimum of the fragment*, it gives the minimum of the target function when combined with *A*. If three is the *fragment of the minimum*, it gives the minimum of the target function when it is combined with *A and* continued on to *B*.

the peptide in the global minimum of the peptide. B is unknown at this stage of the buildup, but we can nevertheless use it in this analysis. Different conformations of the peptide are possible depending upon which partial peptide i is used to concatenate to B. Their target functions are,

$$\Phi_i = \Phi_{A_i} + \Phi_B + E_{A_iB}, \quad i = 1,2,\ldots \qquad (3)$$

where Φ_i is the target function of the full peptide when it is constructed with the ith fragment, and E_{A_iB} is the energy due to interaction between the two segments. If this is zero, the overall target function is exactly additive and the system is a candidate for classical dynamic programming. In this case the minimum among the Φ_{A_i}'s is determined. The corresponding partial peptide is retained and all others ignored.

Nonadditivity can occur when the E_{A_iB}'s are not zero and different for different partial peptides. To explain the cutoff parameter, assume that M (one of the partial peptides) yields the global minimum of the peptide when concatenated with B. We call M the *fragment of the minimum*. Let M' be the partial peptide that corresponds to the minimum among Φ_{A_i}'s. We call this the *minimum of the fragment*. If the target function is nonadditive, M and M' may not be the same.

The problem for EDP is that at this buildup stage only Φ_{A_M} and $\Phi_{A_{M'}}$ are known; Φ_M and $\Phi_{M'}$ are not known. M has to be picked even though $\Phi_{A_{M'}} < \Phi_{A_M}$. It is possible to get an upper bound on the cutoff parameter by realizing that $\Phi_M < \Phi_{M'}$ if and only if $\Phi_{A_M} + E_{A_MB} < \Phi_{A_{M'}} + E_{A_{M'}B}$. Rearranging the terms, we get $\Phi_M < \Phi_{M'}$ if and only if $\Phi_{A_M} < \Phi_{A_{M'}} + E_{A_{M'}B} - E_{A_MB}$. This then defines an upper bound of the cutoff parameter as

$$C = \max\| E_{A_{M'}B} - E_{A_MB} \|$$

since $\Phi_{A_{M'}} > \Phi_{A_M} + C$ always implies that $\Phi_{M'} > \Phi_M$.[26] The maximum is taken over all possible M and M'.

While the above explains cutoff at a molecular level, evaluating the cutoff from such an analysis is difficult or impossible. In this initial study of the algorithm, we chose a small cutoff, taking advantage of the near additivity in the problem, and set it arbitrarily to 0.2 kcal/mol. However, it should be possible to gain increasing confidence in the cutoff parameter by iterating over it in the following manner.

After fully building the peptide (i.e., after ascertaining the global minimum of the peptide), follow its buildup through the stages. At each stage, we consider the conformation of the fragment when it is part of the globally optimum conformation, and the conformation when it is locally optimum. If they are not the same, we compute the difference between the target function of the former conformation, and the cutoff boundary. If the margin is small at one (or more) of the buildup stages, increase the cutoff parameter and rebuild the peptide using the new set of cutoffs. If this leads to a better global minimum, analyze the acceptance margins of this new minimum and change the cutoffs where necessary. Repeat these iterations until the new global minimum found is the same as the one in the previous iteration or until the acceptance margins of the global minimum are comfortably large. For a system that is close to additive, this should happen after only a small number of iterations.

The target function is obviously central to the physics of the problem. In this work we concentrated on the concatenation algorithm and have therefore not elaborated on the target function. However, dramatic improvement can be obtained in the performance of the algorithm by making the calculation of the target function more efficient. Specifically, the target function requires an energy minimization to obviate any steric clashes before it can be applied. Consequently, the algorithm spends an overwhelming majority of its time in the minimization routine. If this is obviated, a large speedup will be achieved.

Let the peptide have N residues and let the ith residue have S_i possible conformations. The num-

ber of evaluations of the target function required by the algorithm scales with N and S_i as

$$\Omega_{\text{eval}} = O\{S_1S_2 + (1 + \alpha_1(C)S_1)S_2S_3 + (1 + \alpha_2(C)S_2(1 + \alpha_1(C)S_1)S_3S_4 + \cdots\}. \quad (4)$$

Here, α's are functions of the cutoff C and go to 0 when $C = 0$ and $(1 - 1/S_i)$ when $C = \infty$. At intermediate values, it is the fraction of the number of previous partial peptides that are within a cutoff of its minimum. When the cutoff parameter remains small, the α's are close to 0 and the number of evaluations of the target function in extended dynamic programming reduces to the same order as classical dynamic programming.

$$\Omega_{\text{eval}} = O(S_1S_2 + S_2S_3 + S_3S_4 + \cdots).$$

The justification for a small cutoff depends heavily on the details of the system. The more important the interactions within the peptide, the less successful a small cutoff is likely to be. Our algorithm takes about 3 days of CPU time on a Silicon Graphics Indigo2 for a hexapeptide with ~ 10 possible positions for each residue. In the mapping done in both the cases presented here, that is the order of magnitude of the problem.

It is worth noting that when the cutoff is infinite (so that the α's are ~ 1), the performance of the algorithm given by eq. (4) is *worse* than an exhaustive search. This is because using an infinite cutoff is the same as building each fragment of the peptide by exhaustive search. The total effort then is the sum of the number of computations required to build the 2-, 3-... N-mers by exhaustive search. This underscores the need to use small cutoffs in the algorithm.

The target function and mapping of the system are intimately related. If the target function is not very sensitive to changes in the structure, the map should be coarse enough to keep the difference in free energies of the adjacent structures detectable. Also, since a minimization precedes the calculation of the target function, the grid points selected for mapping should be sufficiently far apart that two or more points do not converge to the same structure upon minimization. Efficiency requires that mapping give rise to the minimum number of possibilities at each residue. So we have to take advantage of the above two features to map as coarsely as possible. On the other hand, too coarse a mapping is undesirable owing to the possibility of missing conformations that are potentially part of the global minimum. For example, the quality of structures we predicted for the MHC system is not as good as that for the protease system. This is a reflection of the greater coarseness in the MHC map than in the protease map. This trade-off must always be kept in mind when deciding on the density of mapping.

Two extensions of the method are worth mentioning. One is the determination of peptide structure when the structure of the binding site is unknown but obtainable by, say, homologous extension. The main distinction from the case presented here is that searches for the conformation of nonconserved side chains will need to be included in construction of the map. The other is peptide design, which necessitates moving the target function closer to a true thermodynamic free energy function by including the free energy of the free peptide.

Acknowledgments

We thank Benjamin L. King and Osman U. Sezerman for supplying us with the mapping data for the protease and the MHC systems, respectively. This work was supported by Grant AI30535 from the NIAID of the NIH.

References

1. M. Mezei and D. Beveridge, *Ann. NY Acad. Sci. USA*, **494**, 1 (1986).
2. B. Rao, R. Tilton, and U. Singh, *J. Am. Chem. Soc.*, **114**, 4447 (1992).
3. P. A. Kollman, *Curr. Opin. Struct. Biol.*, **4**, 240 (1994).
4. M. Connoly, *Biopolymers*, **25**, 1229 (1986).
5. B. K. Shoichet and I. D. Kuntz, *J. Mol. Biol.*, **221**, 327 (1991).
6. D. J. Bacon and J. Moult, *J. Mol. Biol.*, **225**, 849 (1992).
7. H. Wang, *J. Comp. Chem*, **12**, 746 (1991).
8. J. Novotny, R. Bruccoleri, and F. Saul, *Biochemistry*, **28**, 4735 (1989).
9. K. Smith and B. Honig, *Proteins*, **18**, 119 (1994).
10. S. Vajda, Z. Weng, R. Rosenfeld, and C. DeLisi, *Biochemistry*, **33**, 13977 (1994).
11. P. J. Goodford, *J. Med. Chem.*, **28**, 849 (1985).
12. B. L. King, S. Vajda, and C. DeLisi, submitted for publication, 1995.
13. A. Caflisch, A. Miranker, and M. Karplus, *J. Med. Chem.*, **36**, 2142 (1993).
14. R. Rosenfeld, Q. Zheng, S. Vajda, and C. DeLisi, *J. Mol. Biol.*, **234**, 515 (1993).
15. D. A. Bobbyer, P. J. Goodford, P. M. McWhinnie, and R. C. Wade, *J. Med. Chem.*, **32**, 1083 (1989).

16. S. H. Rotstein and M. A. Murcko, *J. Med. Chem.*, **36**, 1700 (1993).
17. U. Sezerman, S. Vajda, and C. DeLisi, submitted for publication, 1995.
18. R. C. Brower, G. Vasmatzis, M. Silverman, and C. DeLisi, *Biopolymers*, **33**, 329 (1993).
19. R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1975.
20. S. Vajda and C. DeLisi, *Biopolymers*, **29**, 1755 (1990).
21. A. Adamson, *Physical Chemistry of Surfaces*, Wiley, New York, 1976.
22. A. Nicholls, K. Sharp, and B. Honig, *Proteins*, **11**, 281 (1991).
23. M. Miller, J. Schneider, B. K. Satyanarayana, M. V. Toth, G. R. Marshall, L. Clawson, L. Selk, S. B. Kent, and A. Wlodawer, *Science*, **246**, 1149 (1989).
24. D. R. Madden, D. N. Garboczi, and D. C. Wiley, *Cell*, **75**, 693 (1993).
25. R. E. Bruccoleri and J. Novotny, *Immunomethods*, **1**, 96 (1992).
26. S. Vajda and C. Delisi, *The Protein Folding Problem and Tertiary Structure Prediction*, K. Merz and S. Le Grand, Eds., Birkhauser Boston, 1994, p. 411.